THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 59, No. 7, September 1980
Printed in U.S.A.

Address Assignment for a Time-Frequency-Coded, Spread-Spectrum System

By G. EINARSSON

(Manuscript received September 27, 1979)

In a multiple-access, spread-spectrum system, the messages intended for a particular user are distinguished by a specific signal pattern called the address. Here we consider such a system where an address is a sequence of L tones, chosen from Q possible frequencies. It can be described as a pattern in an $L \times Q$ time-frequency matrix. We study the problem of assigning addresses to the user in such a system using an algebraic approach which provides Q distinct addresses that guarantee minimum interference among Q or fewer system users. Both a synchronous and a nonsynchronous situation are considered. In the latter case, we have derived an address assignment that prevents interference from time-shifted signals from other users. We evaluate the performance of the system by studying the message error probability caused by interference among users and present upper bounds that give the maximum number of simultaneous users the system can accommodate at a certain error probability.

I. INTRODUCTION

In multiple-access, spread-spectrum communication, each user has access to the entire system bandwidth.[1,2] One way of distinguishing the signals from different users is to give each user an address consisting of a fixed pattern in time and frequency. The information to be transmitted is modulated or coded onto the address. The receiver detects the appropriate address and decodes the message. This technique is often referred to as random-access discrete address (RADA) or code-division multiple access (CDMA).

This paper deals with such a system proposed by A. J. Viterbi[3] for multiple-access satellite communication by mobile users. It has been

studied for mobile radio communication by D. J. Goodman et al.[4] A somewhat similar system, utilizing a different modulation technique, has been proposed by G. R. Cooper and R. W. Nettleton[5] and has been investigated by P. S. Henry.[6]

We consider the problem of generating the addresses assigned to the individual users. By means of an algebraic method, we derive a set of addresses that guarantee minimum interference.

II. SYSTEM DESCRIPTION

We consider a time-frequency coded system where the transmitted signal from each user is a sequence of L tones, chosen from Q possible frequencies. The transmission time T for the code words is assumed to be the same for all users. Each code word conveys $\log_2 Q$ bits of information resulting in an information rate $R = (\log_2 Q)/T$ b/s. The total system bandwidth W is divided into Q subchannels each of width W/Q and the message time T is divided into L time slots each of duration T/L. The possible signals employed in the system are characterized by tbe matrix shown in Fig. 1 having Q rows representing frequency channels and L columns representing time slots. The elements of the matrix, usually called chips, have a time-bandwidth product equal to WT/QL. It is assumed to be large enough so that signals located in different chips are orthogonal, i.e., they will not interfere with each other.

Each user is assigned an address occupying L chips, one in each column of the matrix. Figure 1 shows an example of such an address. The message can modulate the address in different ways. In Ref. 3, it is done by shifting the address vertically in the matrix. With each shift corresponding to a transmitted message, there is a total of Q possible messages. The transmitted code word shown in Fig. 1 is obtained by shifting the address cyclically three steps.

It is convenient to describe the modulation procedure in algebraic terms. Let the address of user m be denoted by a vector.

$$\mathbf{a}_m = (a_{m1}, a_{m2}, \cdots, a_{mL}), \tag{1}$$

where $a_{mi} \in GF(Q)$ represents the frequency channel occupied by the address at time slot i. The symbol $GF(Q)$ denotes the finite field (Galois field) of Q elements. A summary of some basic features of finite fields can be found in Appendix A.

For Q a prime number, the rule of addition in $GF(Q)$ is ordinary addition modulo Q. A transmitted sequence $\mathbf{y}$ formed by the operation

$$\mathbf{y}_m = \mathbf{a}_m + x_m \cdot \mathbf{1} \tag{2}$$

will therefore correspond to the address $\mathbf{a}_m$ shifted cyclically x_m steps in the signal matrix. The notation $\mathbf{1}$ is used for the all-one vector $\mathbf{1} =$

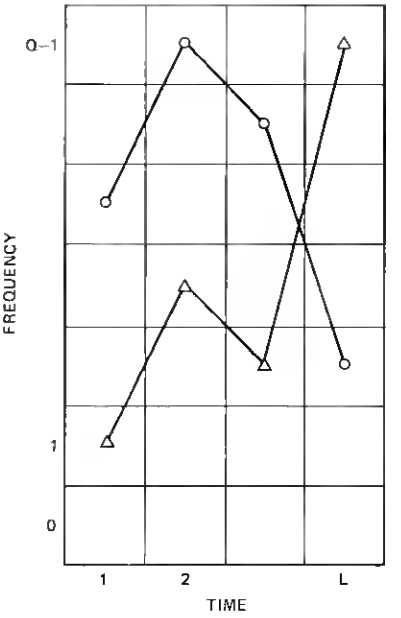

Fig. 1—Signal matrix showing an address (△——△) and a transmitted signal (○——○).

(1, 1, ⋯ , 1). When Q is not a prime number, the rule of addition is different (see Appendix A). It is still a simple and well-defined operation transforming $\mathbf{a}_m$ into $\mathbf{y}_m$.

The receiver scans the received signal matrix and determines which chips are active. It then subtracts the address $\mathbf{a}_m$ resulting in the message $x_m \cdot \mathbf{1}$ appearing as a complete row in the matrix.

III. ADDRESS ASSIGNMENT

For the moment we consider a synchronous system, which means that the signals from all users are aligned in time. Two transmitted signals will interfere if and only if they occupy the same time-frequency chip in the signal matrix. This happens if the corresponding vectors $\mathbf{y}$ have coinciding symbols.

We propose the following way of generating a set of Q addresses. Let

$$\mathbf{a}_m = (\gamma_m, \gamma_m\beta, \gamma_m\beta^2, \cdots, \gamma_m\beta^{L-1}), \tag{3}$$

where γ_m is an element of $GF(Q)$ assigned to user m and β is a fixed primitive element of $GF(Q)$.

To see to what extent transmitted signals can interfere with each other, let $\mathbf{y}_1$ and $\mathbf{y}_2$ denote two sequences generated by two different addresses $\mathbf{a}_1$ and $\mathbf{a}_2$, respectively.

$$\left.\begin{aligned}\mathbf{y}_1 &= \mathbf{a}_1 + x_1 \cdot \mathbf{1} \\ \mathbf{y}_2 &= \mathbf{a}_2 + x_2 \cdot \mathbf{1}\end{aligned}\right\}. \tag{4}$$

The symbols in position i are

$$\left.\begin{aligned}y_{1i} &= a_{1i} + x_1 = \gamma_1 \beta^{i-1} + x_1 \\ y_{2i} &= a_{2i} + x_2 = \gamma_2 \beta^{i-1} + x_2\end{aligned}\right\}. \tag{5}$$

Assume $y_{1i} = y_{2i}$, in which case

$$(\gamma_1 - \gamma_2)\beta^{i-1} + x_1 - x_2 = 0. \tag{6}$$

Consider next the symbols y_{1j} and y_{2j} at position $j \neq i$. We have

$$y_{1j} - y_{2j} = (\gamma_1 - \gamma_2)\beta^{j-1} + x_1 - x_2. \tag{7}$$

Substitution of (6) gives

$$y_{1j} - y_{2j} = (\gamma_1 - \gamma_2)(\beta^{j-1} - \beta^{i-1}). \tag{8}$$

According to our assumptions, $\mathbf{a}_1 \neq \mathbf{a}_2$, which implies that $(\gamma_1 - \gamma_2) \neq 0$. The factor $(\beta^{j-1} - \beta^{i-1})$ is different from zero as long as j and i are less than or equal to $Q - 1$, which is the order of β. (By definition, if β is a primitive element of $GF(Q)$ all powers of β up to $Q - 1$ will be different. See Appendix A.)

Since (8) applies to an arbitrary $j \neq i$, we have shown that, with appropriate choice of β, two transmitted vectors with different addresses will coincide in at most one chip. Notice that this is true independent of the information transmitted. From (4) and (5), it follows that messages exist that cause coincidence at position i for any choice of the address symbols a_{1i} and a_{2i}. Since the method proposed can have no more than one coincidence, it is optimum in terms of the amount of interference between users. The maximum number of addresses obtained is equal to Q, and the maximum possible value of L is equal to $Q - 1$.

Example 1:

Let $Q = 7$ and $L = 4$.

Since 7 is a prime number, the modulo Q arithmetic presented in Appendix A applies. With $\beta = 3$, which is a primitive element (i.e., of order 6), the following set of addresses are obtained from (3) with $\gamma_m = m$ ($m = 0, 1, 2, \cdots, Q - 1$).

$\mathbf{a} =$	0	0	0	0
	1	3	2	6
	2	6	4	5
	3	2	6	4
	4	5	1	3
	5	1	3	2
	6	4	5	1.

The address $\mathbf{a}_1 = (1, 3, 2, 6)$ in conjunction with message $x = 3$ will result in $\mathbf{y}_1 = (1, 3, 2, 6) + (3, 3, 3, 3) = (4, 6, 5, 2)$, which is depicted in Fig. 1.

Notice that the address set shows each symbol of $GF(Q)$ once and only once in each column, which means that the Q addresses completely fill the signal matrix without interference.

Example 2:

Consider $Q = 8$ and $L = 4$, where Q is of the form $Q = 2^K$. Such a Q is advantageous because each message conveys K bits of information and because the operations in $GF(2^K)$ are readily implemented with binary logic. We use the octal representation introduced in Appendix A with (101) = 5, etc. The element $\beta = (010) = 2$ is primitive and has the following sequence of powers $\beta^0, \beta^1, \cdots, \beta^6 = 1, 2, 4, 3, 6, 7, 5$. See Appendix A for details.

Equation (3) then gives the following set of addresses

$$\mathbf{a} = \begin{matrix} 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 3 \\ 2 & 4 & 3 & 6 \\ 3 & 6 & 7 & 5 \\ 4 & 3 & 6 & 7 \\ 5 & 1 & 2 & 4 \\ 6 & 7 & 5 & 1 \\ 7 & 5 & 1 & 2. \end{matrix}$$

The addition in (2) should now be performed according to the rules of $GF(8)$. As shown in Appendix A, this is accomplished by bit-by-bit addition (exclusive OR) of the three-digit binary numbers representing the elements of $GF(8)$, i.e., $3 + 2 = (011) + (010) = (001) = 1$. The address $\mathbf{a}_1 = (1, 2, 4, 3)$ and message $x = 2$ will result in $\mathbf{y}_1 = (3, 0, 6, 1)$.

The change-in-addition rule does not affect the communication capability of the system. The receiver detects the chips separately and decodes the message by subtracting the address from the received sequence, i.e., performs the inverse operation of (2). This works equally well if Q is a prime number or not.

IV. NONSYNCHRONOUS SYSTEM

In practice, it may be difficult to achieve synchronization between all users in the system. We consider a situation where not only a transmitted sequence itself but also a time-shifted version of it can cause interference to another user. The address assignment described in Section III is not well suited for this nonsynchronous case because the addresses generated by (3) are cyclic shifts or shortened cyclic shifts of each other.

As an example, consider the addresses $\mathbf{a}_1 = (1, 3, 2, 6)$ and $\mathbf{a}_3 = (3, 2, 6, 4)$ of Example 1. A shift of one element to the left in $\mathbf{a}_1$ causes three symbols to coincide with $\mathbf{a}_3$. This has the effect that time-shifted versions of transmitted sequences from different users have a high tendency to coincide and cause interference. In the special case when $L = Q - 1$, the addresses are true cyclic shifts of each other. This means that any vector $\mathbf{y}$ from any user will then be transformed into a message for another user by a cyclic shift rendering the modulation scheme completely useless without synchronization.

The cyclic property of the address set generated by (3) is due to the fact that the addresses form a shortened cyclic code (with one information symbol γ and $L - 1$ check symbols). It is a (shortened) Reed-Solomon code. The use of such codes to generate sequences of minimal interference for spread spectrum communication has been suggested in Refs. 7 and 8.

One easy way to obtain a system resistant to nonsynchronous interference is to let the transmitted sequence $\mathbf{y}$ be formed by the new rule

$$\mathbf{y}_m = x_m \cdot \boldsymbol{\beta} + \gamma_m \cdot \mathbf{1}, \tag{9}$$

where $\boldsymbol{\beta} = (1, \beta, \beta^2, \cdots, \beta^{L-1})$, $\gamma_m \in GF(Q)$ represents the address of user m, and $x_m \in GF(Q)$, as before, is the message symbol.

Let $\mathbf{y}^1$ and $\mathbf{y}^2$ denote the transmitted sequences from two users

$$\begin{aligned} \mathbf{y}^1 &= x_1 \cdot \boldsymbol{\beta} + \gamma_1 \cdot \mathbf{1} \\ y^2 &= x_2 \cdot \boldsymbol{\beta} + \gamma_2 \cdot 1. \end{aligned} \tag{10}$$

We consider the interference between $\mathbf{y}^1$ and $\mathbf{y}^2$ shifted k steps to the left. The symbol in position i of $\mathbf{y}^1$ is

$$y_i^1 = x_1 \beta^{i-1} + \gamma_1. \tag{11}$$

The symbol in $\mathbf{y}^2$ that can interfere with y_i^1 is

$$y_{i+k}^2 = x_2 \beta^{i+k-1} + \gamma_2. \tag{12}$$

The condition for interference is

$$y_i^1 - y_{i+k}^2 = x_1 \beta^{i-1} - x_2 \beta^{i+k-1} + \gamma_1 - \gamma_2 = 0. \tag{13}$$

Two symbols y_j^1 and y_{j+k}^2 at an arbitrary position $j \neq i$ have the difference

$$y_j^1 - y_{j+k}^2 = x_1 \beta^{j-1} - x_2 \beta^{j+k-1} + \gamma_1 - \gamma_2. \tag{14}$$

Combining (13) and (14) gives

$$\begin{aligned} y_j^1 - y_{j+k}^2 &= -\beta^{j-i}(\gamma_1 - \gamma_2) + \gamma_1 - \gamma_2 \\ &= (1 - \beta^{j-i})(\gamma_1 - \gamma_2) \neq 0. \end{aligned} \tag{15}$$

Equation (15) shows that two sequences $\mathbf{y}$ from different users will not coincide in more than one position for any cyclic shift between the two sequences. In a nonsynchronous situation, parts of two consecutive code words from an interferer can fall into the message time T for a receiver. If these code words are identical, we have the case of cyclic-shifted sequences analyzed above, and only one coincidence is possible. When the interfering messages are different, each of them can produce at most one coincidence. Since this is the best that can be achieved by any address selection, procedure (9) is optimum in the sense of generating least interference between nonsynchronous users.

The fundamental difference between (2) and (9) is that, in the former case, the Q *addresses* constitute a shortened cyclic code, in the latter case the Q possible *messages* belonging to a particular user form a cyclic set.

Example 3:

Let $Q = 8$ and $L = 4$.

The address set according to (9) consists of the vectors $\mathbf{a}_1 = (0, 0, 0, 0)$, $\mathbf{a}_2 = (1, 1, 1, 1)$, etc.

With $\beta = 2$, the vector $\boldsymbol{\beta}$ is $(\beta^0, \beta^1, \beta^2, \beta^3) = (1, 2, 4, 3)$. The multiplication of $\boldsymbol{\beta}$ by x is most easily carried out by expressing x as a power of β. For example, $x = 3$ is equal to β^3 and $\beta^3\boldsymbol{\beta} = (\beta^3, \beta^4, \beta^5, \beta^6) = (3, 6, 7, 5)$.

The addition of $\mathbf{a}$ to form the transmitted sequence $\mathbf{y}$ is the binary number addition previously used in Example 2. The address $\mathbf{a}_3 = (2, 2, 2, 2)$ will give $\mathbf{y}_3 = (3, 6, 7, 5) + (2, 2, 2, 2) = (1, 4, 5, 7)$, and so on.

In contrast to the previous case, a system operating according to (9) will not directly reveal the transmitted information by subtraction of the address from the received sequence. A receiver detects each chip in the signal matrix and suhtracts the address γ from all chips. It then multiplies the first column by $\beta^0 = 1$, the next by β^{-1}, the third by β^{-2}, and so on. This transforms the vector $x \cdot \boldsymbol{\beta}$ into $x \cdot \mathbf{1}$, which will show up as a complete row in the matrix at position x.

V. ERROR PROBABILITY

Consider a situation where there is no noise in the system and the only source of errors is interference from other users. Also assume that all signals transmitted in the system will be received by any user creating additional entries in his signal matrix.

A decoded message shows up as a complete row of filled chips in the signal matrix. For an error to occur the interfering signals must combine into one or more complete rows located elsewhere in the matrix.

One way of estimating the error probability is to use a random

coding argument. The addresses are assigned randomly, and the average error probability is calculated over all possible assignments. This is equivalent to the assumption that all transmitted chips are independent and have equal probability $1/Q$ of taking any of the Q possible values.

The probability of $M-1$ interfering users producing an erroneous complete row of length L at a specific location in the signal matrix is then[3,4]

$$P_R = [1-(1-1/Q)^{M-1}]^L. \tag{16}$$

The probability that a false row appears in at least one of the $Q-1$ positions not corresponding to the transmitted message is bounded by the union bound

$$P_W \le P_{W1} = (Q-1)[1-(1-1/Q)^{M-1}]^L. \tag{17}$$

The expression (17) constitutes an upper bound to the message error probability when addresses are chosen randomly. An expression for the bit error probability can be obtained from it in the way shown in Ref. 3.

Bounds on the error probability for the specific address assignment proposed in this report are derived in Appendix B. For a synchronous system, a simple combinatorial argument gives

$$P_W \le P_{W2} = (Q-1)\frac{(M-1)(M-2)\cdots(M-L)}{Q^L}. \quad (18)=(22)$$

By studying the probability of a chip being filled conditional on previous chips being occupied, an alternative upper bound on P_W is obtained:

$$P_W < P_{W3} = (Q-1)\prod_{k=0}^{L-1}\left[1-\left(1-\frac{1}{Q-k}\right)^{M-k-1}\right]. \quad (19)=(30)$$

The right-hand side of (19) is always less than (17). The word error probability bounds (18) and (19) are plotted in Fig. 2 for $Q=32$ and $L=12$ together with (17). The diagram shows that (18) is a better bound than (19) at low error probability, while the opposite is true at higher values. Notice that when $M \le L$ the error probabilities (18) and (19) are equal to zero because tones from $L-1$ interfering users will not be able to combine into a complete row.

For a nonsynchronous system, the calculation of error probability is more complicated than in the synchronous case. In Appendix B the error probability for a system with signals generated according to (9) is studied and the following expression derived:

$$P_{W4} = (Q-1)\left\{\prod_{k=0}^{L/2-1}\left[1-\left(1-\frac{1}{Q-k}\right)^{M-k-1}\right]\right\}^2, \quad (20)=(32)$$

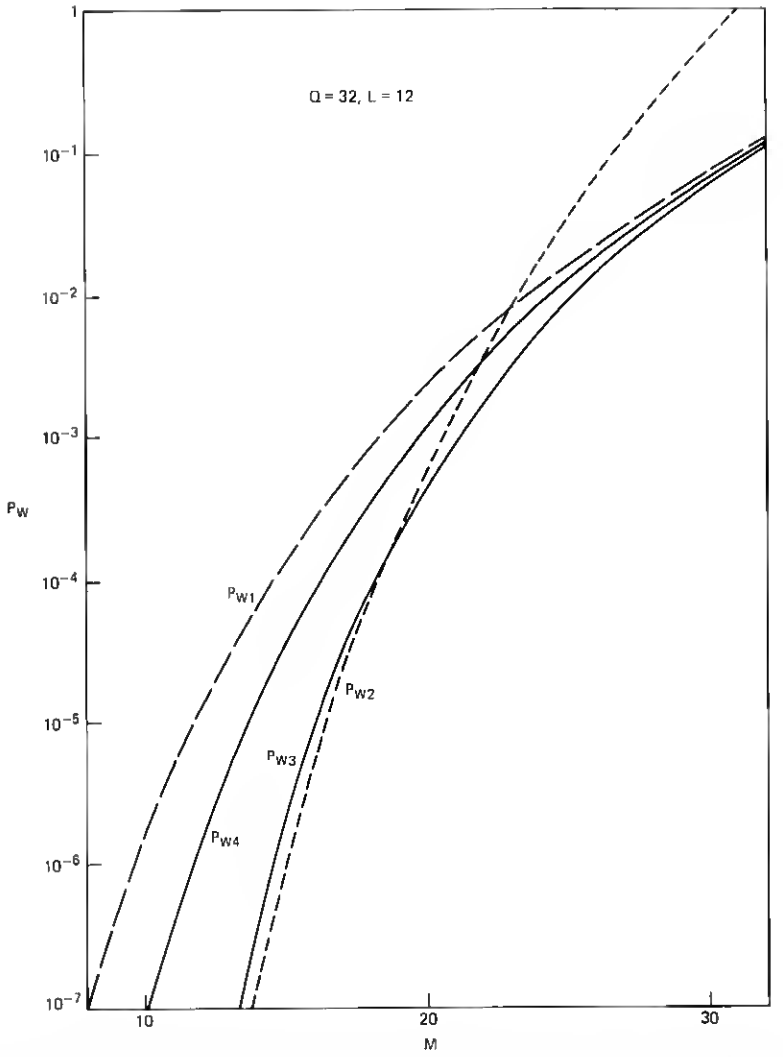

Fig. 2—Word error probability as functions of the number of users M for a system with $Q = 32$ and $L = 12$. P_{W1} is the average probability for random coding. P_{W2} and P_{W3} are upper bounds for a synchronous system with address assignment according to eqs. (2) or (9). P_{W4} represents a nonsynchronous system with the address assignment of eq. (9).

where L is assumed to be even. P_{W4} is greater than the synchronous bound (19) but less than the random coding bound (17). It is plotted for $Q = 32$ and $L = 12$ in Fig. 2.

It has been assumed that the receiver cannot be certain of what message was transmitted when more than one complete row shows up

in the decoded signal matrix. A more complicated receiver could investigate which rows are due to interference and which is due to the wanted message by, in principle, decoding the messages from all users. Such a scheme of complete detection would reduce the probability of error but seems hard to analyze.

VI. ACKNOWLEDGMENTS

I wish to thank David Goodman for introducing me to the problem and for valuable comments on the manuscript and Barry Haskell for suggesting the formula (19) for the error probability.

APPENDIX A

Algebra of Finite Fields

A1. Definitions

This appendix presents a few elementary concepts of algebra for finite fields needed to understand the address assignment procedures in Sections II and III. For a more extensive and rigorous account, see Albert[9] or Chapter 4 of Berlekamp.[10]

A finite field $GF(Q)$ is a set of Q elements with rules for addition (and subtraction) and multiplication (and division) consistently defined. As a consequence, a finite field has a zero element and a unit element, both being unique. The zero element has the property $a + 0 = a$ and for the unit element $a \cdot 1 = a$ for all $a \in GF(Q)$.

We denote the elements of $GF(Q)$ by the integers 0, 1, 2, $\cdots$, $Q - 1$ with 0 and 1 the zero and unit element, respectively. It should be realized that this notation is arbitrary and that the rules of addition and multiplication have not yet been specified.

A fundamental result of higher algebra is that there exist finite fields only for Q equal to a prime or the power of a prime number p, i.e., $Q = p^K$. This means that $Q = 2, 3, 4, 5, 7, 8, 9, 11, 13$, etc. are permissible but there is no finite field with, for instance, 10 elements. All fields with Q elements are isomorphic which means that they differ only in the way the elements are named. The field of Q elements is called the Galois field after the French mathematician Évariste Galois, which explains the notation $GF(Q)$.

A2. Q equal to a prime number

When Q is equal to a prime number, the rules of addition and multiplication in $GF(Q)$ are defined by modulo Q arithmetic. This means that the sum or product between two elements is defined as this operation in the usual algebra of integer numbers with the results reduced modulo Q (i.e., equal to the remainder after dividing by Q). Let $Q = 7$. We then have, for example: $2 \cdot 3 = 6$, $1 + 4 = 5$, $4 \cdot 3 = 5$ ($=12 \bmod 7$), $2 + 5 = 0$ ($=7 \bmod 7$).

A nonzero element $a \in GF(Q)$ is said to be of (multiplicative) order N if N is the lowest nonzero integer such that $a^N = 1$. Since a^j is equal to a nonzero element and there are $Q - 1$ such elements in the field, N must be less than or equal to $Q - 1$. An element with $N = Q - 1$ is called a primitive element. Examples: In $GF(7)$, the element $a = 2$ has the powers $a^0, a^1, a^2, \cdots = 1, 2, 4, 1, 2, \cdots$. The order of $a = 2$ is thus $N = 3$. The element $a = 3$ has the powers $a^0, a^1, a^2, \cdots = 1, 3, 2, 6, 4, 5, 1, 3, \cdots$, which shows that $a = 3$ is a primitive element, i.e., of order $N = 6$. It is clear from the examples that the powers of a primitive element are all the nonzero elements of a finite field.

A3. Q equal to the power of a prime number

When Q is a prime number, the rules for addition and multiplication are related to ordinary real number algebra in a simple way. When Q is a power of a prime number, things are a little more complicated. Consider the case $p = 2$, which is important since the operations in the field can then be instrumented by binary circuitry.

To define addition in a field with $Q = p^K$, the elements of the field are represented as p-ary numbers or vectors of length K. As an example, for $Q = 2^3 = 8$ the elements are expressed as three-digit binary numbers: $1 = 001$, $3 = 011$, $4 = 100$, etc. which makes 0, 1, $\cdots$, 7 the octal representation of these numbers.

Addition is now defined as mod p addition of the components. For $p = 2$, this is the binary addition: $1 + 0 = 1$ and $1 + 1 = 0$, which gives $3 + 1 = 011 + 001 = 010 = 2$, $5 + 4 = 101 + 100 = 001 = 1$, etc.

To specify multiplication in $GF(p^K)$, the K-tuplets are transformed into polynomials in z of degree $K - 1$ by letting the first digit be the coefficient of z^{K-1}, the second digit the coefficient of z^{K-2}, etc. The triplet (111) corresponds to $z^2 + z + 1$, (011) to $z + 1$, and so on. Addition and multiplication of polynomials are defined as in ordinary algebra using the mod-p rule for the coefficients. The multiplication rule of $GF(p^K)$ is polynomial multiplication modulo an irreducible polynomial $P(z)$ of degree K. A polynomial is irreducible if it is not possible to factor it into a product of polynomials of lower degree. It thus has the features of a prime number in the algebra of polynomials.

As an example, consider $p = 2$ and $K = 3$. The polynomial $P(z) = z^3 + z + 1$ is irreducible. The multiplication of $5 = (101)$ and $3 = (011)$ gives $(z^2 + 1)(z + 1) = (z^2) = (z^3 + z^2 + z + 1) \bmod P(z)$. We thus have $5 \cdot 3 = (100) = 4$ in $GF(8)$.

The element $b = (z) = (010)$ is primitive, having the table of powers shown in Table I. Multiplication in $GF(p^K)$ is most easily performed by using such a list of the nonzero elements expressed as the power of a primitive element. For instance, $5 = b^6$ and $3 = b^3$ giving $5 \cdot 3 = b^{6+3} = b^9 = b^7 \cdot b^2 = b^2 = 4$.

Table I—Powers of b = 2

		Binary	Octal
b^0	1	001	1
b^1	z	010	2
b^2	z^2	100	4
b^3	$z^3 = z + 1$	011	3
b^4	$z^2 + z$	110	6
b^5	$z^3 + z^2 = z^2 + z + 1$	111	7
b^6	$z^3 + z^2 + z = z^2 + 1$	101	5
b^7	$z^3 + z = 1$	001	1

APPENDIX B

Error Probability

An error is assumed to occur when interfering signals combine in such a way that one or more erroneous messages occur in the signal matrix. We will upper-bound the probability P_R that a particular sequence, say $\mathbf{y}_1$ of user 1, will be formed by the $M - 1$ interfering signals from the other users in the system.

For a synchronous system, we have proved that any signal can coincide with any other signal in at most one chip, which means that at least L interfering signals are needed to make an error. Moreover, for any user one and only one of its Q messages can occupy a specified chip. The number of ways in which L out of $M - 1$ signals can combine into a specific pattern of L chips is therefore $(M - 1)(M - 2) \cdots (M - L)$. For any such combination, the remaining $M - L - 1$ signals can take Q^{M-L-1} values. The expressions

$$(M - 1)(M - 2) \cdots (M - L)Q^{M-L-1} \tag{21}$$

is therefore an upper bound to the number of combinations of $M - 1$ signals that will result in an erroneous message. It is a bound and not an exact expression since combinations with more than L signals contributing to the error are counted more than once. By dividing (21) by Q^{M-1}, the total number of interfering signal combinations, the following bound for P_R is obtained:

$$P_R \leq \frac{(M - 1)(M - 2) \cdots (M - L)}{Q^L}. \tag{22}$$

A bound that is better than (22) at higher error probabilities is obtained in the following way. Let I_k denote the number of interfering signals occupying chip number k, $k = 1, 2, \cdots, L$, of the fixed pattern $\mathbf{y}_1$. The probability P_R of $\mathbf{y}_1$ occurring in the signal matrix is then a product of conditional probabilities

$$P_R = P(I_1 \neq 0)P(I_2 \neq 0 | I_1 \neq 0) \cdots P(I_L \neq 0 | I_1 \neq 0, I_2 \neq 0, \cdots I_{L-1} \neq 0). \tag{23}$$

The need for conditional probabilities in (22) is due to the fact that the address coding introduces dependence between transmitted symbols from a certain user.

The first term in (23) is

$$P(I_1 \neq 0) = 1 - \left(1 - \frac{1}{Q}\right)^{M-1}, \tag{24}$$

where $[1 - (1/Q)]$ is the probability of the chip not being occupied by a single interfering signal.

The first two terms of (23) can be expanded into

$$P(I_1 \neq 0)P(I_2 \neq 0 | I_1 \neq 0) = \sum_{i=1}^{M-1} P(I_1 = i)P(I_2 \neq 0 | I_1 = i). \tag{25}$$

When $I_1 = i$, there are $M - i - 1$ interfering signals which can contribute to I_2. We therefore have

$$P(I_2 \neq 0 | I_1 = i) = 1 - \left(1 - \frac{1}{Q-1}\right)^{M-i-1}, \tag{26}$$

where $1/(Q - 1)$ is the probability of chip no. 2 being filled by any of the $M - i - 1$ signals not occupying chip number one. The factor $Q - 1$ is due to the fact that these signals are assumed not to occupy chip no. 1, which reduces the number of possible choices from Q to $Q - 1$.

From (26) follows that $P(I_2 \neq 0 | I_1 = i)$ takes its largest value when $i = 1$. We can therefore overbound the right-hand side of (25) by

$$P(I_2 \neq 0 | I_1 = 1) \sum_{i=1}^{M-1} P(I_1 = i) = P(I_2 \neq 0 | I_1 = 1)P(I_1 \neq 0). \tag{27}$$

Combining (24) and (26) for $i = 1$ gives

$$P(I_1 \neq 0)P(I_2 \neq 0 | I_1 \neq 0) < \left[1 - \left(1 - \frac{1}{Q}\right)^{M-1}\right]\left[1 - \left(1 - \frac{1}{Q-1}\right)^{M-2}\right]. \tag{28}$$

The next term in (23) is bounded the same way by replacing

$$P(I_3 \neq 0 | I_1 = i, I_2 = j) = 1 - \left(1 - \frac{1}{Q-2}\right)^{M-i-j-1} \tag{29}$$

by its largest value, which is given by $i = j = 1$. Continuing this reasoning, we arrive at

$$P_R < \prod_{k=0}^{L-1} \left[1 - \left(1 - \frac{1}{Q-k}\right)^{M-k-1}\right]. \tag{30}$$

The analysis above assumes a synchronous system. In the nonsynchronous case, the interfering sequences need not be aligned in time

with the message sequence $\mathbf{y}_1$. For the coding rule (9), we have proved that an interfering user can coincide with $\mathbf{y}_1$ in at most two chips.

The probability of an interferer producing coincidences is a function of the alignment. The probability of two coincidences in a word of length L is

$$P(I = 2) = \frac{k}{Q}\frac{L-k}{Q}, \tag{31}$$

where $k = 0, 1, \cdots, L-1$ represents the misalignment between the two sequences. See Fig. 3. $P(I = 2)$ is zero for $k = 0$, which is the synchronous case and takes it maximum value for $k = L/2$ (L assumed even).

Next consider a situation where L is even and all the $M - 1$ distributing signals are placed in the position $k = L/2$ with respect to $\mathbf{y}_1$. A bound on the error probability for such a system is easily obtained by observing that the interference in chips 1 to $L/2$ and in chips $L/2 + 1$ to L are independent. The probability of at least $L/2$ coincidences is for each part bounded by (30) with L replaced by $L/2$, yielding

$$P_R < \left\{ \prod_{k=0}^{L/2-1} \left[1 - \left(1 - \frac{1}{Q-k} \right)^{M-k-1} \right] \right\}^2. \tag{32}$$

The error probability (32) is calculated for a system where the interferers are most harmful in terms of their probability of producing two coincidences. We believe that it constitutes an upper bound also for a general nonsynchronous situation with arbitrary alignment between the users.

In all cases considered, an upper bound on the probability P_W for the occurrence of at least one false message is obtained from P_R by the union bound

$$P_W \leq (Q-1)P_R. \tag{33}$$

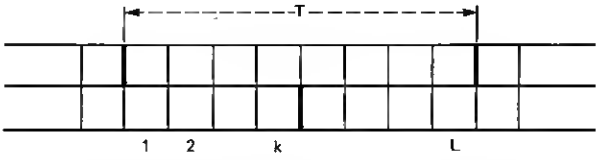

Fig. 3—Two signals of a nonsynchronous system. The displacement k takes integer values $0 \leq k \leq L - 1$.

REFERENCES

1. R. C. Dixon, *Spread Spectrum Systems*, New York: John Wiley, 1976.
2. R. C. Dixon, ed., *Spread Spectrum Techniques*, New York: IEEE Press, 1976.
3. A. J. Viterbi, "A Processing Satellite Transponder for Multiple Access by Low-Rate Mobile Users," Digital Sat. Commun. Conf., Montreal, October 23–25, 1978.
4. D. J. Goodman, P. S. Henry, and V. K. Prabhu, "Frequency-Hopped Multilevel FSK for Mobile Radio," B.S.T.J., this issue, pp. 1257–1275.
5. G. R. Cooper and R. W. Nettleton, "A Spread-Spectrum Technique for High-Capacity Mobile Communications," IEEE Trans. Vehic. Tech., *VT-27*, No. 4 (November 1978), pp. 264–275.
6. P. S. Henry, "Spectrum Efficiency of the Cooper-Nettleton Spread-Spectrum Mobile Radio System," IEEE Trans. Vehic. Tech., *VT-28*, No. 4 (November 1979) 1, pp. 327–332.
7. G. Solomon, "Optimal Frequency Hopping Sequences for Multiple Access," Proc. 1973 Sympo. Spread Spectrum Commun., San Diego, Calif., March 13–16, 1973, Vol. 1, pp. 33–35.
8. D. V. Sarwate and M. B. Pursely, "Hopping Patterns for Frequency-Hopped Multiple-Access Communication," Int. Conf. Commun., ICC78, Toronto, Canada, June 4–7, 1978, pp. 7.4.1–7.4.3.
9. A. A. Albert, "Fundamental Concepts of Higher Algebra," Chicago: University of Chicago Press, 1961.
10. E. R. Berlekamp, "Algebraic Coding Theory," New York; McGraw-Hill, 1968.